Chapter 85
Challenges and Solutions for Addressing Software Security in Agile Software Development:
A Literature Review and Rigor and Relevance Assessment

Ronald Jabangwe
University of Southern Denmark, Odense M, Denmark

Martin S Hubel
University of Southern Denmark, Odense, Denmark

Kati Kuusinen
University of Southern Denmark, Odense, Denmark

Hasan M Alradhi
University of Southern Denmark, Odense M, Denmark

Klaus R Riisom
University of Southern Denmark, Odense, Denmark

Niels Bonde Nielsen
University of Southern Denmark, Odense S, Denmark

ABSTRACT

There has been a surge in the number of software security threats and vulnerabilities in recent times. At the same time, expectations towards software and data security are growing. Thus, there is a need to ensure that security-related tasks are effectively integrated in the software development processes. However, integrating security practices with agile software development is not trivial due to, for instance, differences in process dynamics and the concentration on functional vs non-functional requirements. In this article, the authors present a literature review on the challenges and solutions when adopting security in an agile software development context. Their findings suggest that there are ongoing efforts to integrate security-practices in agile methods, but more research is needed to make the processes more optimized and simpler for developers. A rigor and relevance assessment on primary studies highlights a need for improving the manner in which studies on the topic are performed as well as reported.

DOI: 10.4018/978-1-7998-3016-0.ch085

1. INTRODUCTION

Security is a crucial quality in software. It aims to prevent, mitigate or avoid exploitation of vulnerabilities. If vulnerabilities creep into the software, they provide attackers an opportunity to perform malicious activities and in turn compromise security properties of the software (Lebanidze, 2006). Thus, there is a need to incorporate practices into the development process for mitigating the development of insecure software systems.

Security can be decomposed into properties of availability, confidentiality and integrity (Avizienis et al., 2004). Availability refers to the guarantee that authorized users can access resources provided by the software system (Kissel, 2013). Confidentiality refers to revealing information to authorized users only (Kissel, 2013). Integrity ensures information is not manipulated by anyone other than the authorized users (Kissel, 2013). The ultimate goal of security engineering is to build dependable software systems in the face of malicious users (Anderson, 2010).

There is a widespread industrial adoption and increasing academic interest on agile software engineering. Agile aims to be a lightweight, iterative and incremental approach to software engineering that welcomes changing requirements throughout the software lifecycle (Cockburn and Highsmith, 2001). As the original agile methodologies such as Scrum (Schwaber and Beedle, 2002) do not address security engineering, the integration of agility and security is not straightforward. Moreover, there are some rather inherent differences between the agile philosophy and some security engineering practices. For instance, threat modeling and requirements engineering are crucial parts of security engineering (Anderson, 2010). These practices usually require upfront design and relatively heavy documentation which can contradict with agile methods that focus on the early and continuous delivery of working software (Beck et al., 2001). Thus, it is of interest to researchers and practitioners to identify agile methods and practices that incorporate security activities that have been reported in peer-reviewed articles. The identification of such methods and practices can contribute towards the development of evidence-based approaches to the integration of security engineering and agile software engineering.

In this paper, we compile the processes and practices used to apply security in agile software development. Along with these are the encountered challenges and their corresponding solutions as reported in peer-reviewed articles.

The remainder of the paper is structured as follows. Section 2 summarizes existing related work. Section 3 describes the review steps, and Section 4 provides results of the steps. Section 5 describes the results collected to answer research questions. The results are then discussed in Section 6. Section 7 concludes the paper and provides possible areas of future work.

2. RELATED WORK

A review published in 2013 (Ghani and Yasin, 2013) suggests that eXtreme Programming (XP) needs to be extended to include security practices, roles and guidelines to help build secure systems. The authors argue that XP already supports adoption of security-related practices. They give an example that pair-programming and coding standards are practices that can be useful for enhancing secure coding practices. On the other hand, it can also be argued that poor coding rules and practices can be a source of vulnerabilities (Liaqat et al., 2015).